

# Carbon Charts: Software Requirements Specification

McMaster University

Group 18

Darsi Anandarajah

Justin Licari

Eliad Moosavi

Thomas Mullen

Richard Zhang

December 27, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Project Drivers</b>	<b>1</b>
2.1	Background of the Project Effort . . . . .	1
2.2	Scope . . . . .	1
2.2.1	Benefits . . . . .	2
2.3	The Stakeholders . . . . .	2
2.3.1	IBM Canada Ltd . . . . .	2
2.3.2	Application Developers . . . . .	2
2.3.3	Application Users . . . . .	3
2.4	Naming Conventions and Terminology . . . . .	3
2.5	References . . . . .	4
2.6	Constraints . . . . .	4
2.6.1	Solution Constraints . . . . .	4
2.6.2	Schedule Constraints . . . . .	4
2.6.3	Budget Constraints . . . . .	4
<b>3</b>	<b>Off-The-Shelf Solutions</b>	<b>4</b>
3.1	Relevant Facts and Assumptions . . . . .	5
3.1.1	Facts . . . . .	5
3.1.2	Assumptions . . . . .	5
3.2	Context Diagrams . . . . .	6
<b>4</b>	<b>Functional</b>	<b>6</b>
4.1	IBM and Library Requirements . . . . .	6
4.2	Bar Chart : Stack and Grouped . . . . .	8
4.3	Circle Charts : Pie, Donut and Bubble . . . . .	9
4.3.1	Pie and Donut Charts . . . . .	9
4.3.2	Bubble Charts . . . . .	10
4.4	Line Chart . . . . .	10
4.5	Area Chart . . . . .	11
4.6	Hardware Requirements . . . . .	11
<b>5</b>	<b>Non-Functional</b>	<b>11</b>
5.1	Look and Feel . . . . .	11
5.2	Usability . . . . .	12

5.3	Safety . . . . .	13
5.4	Performance . . . . .	13
5.5	Installability . . . . .	14
5.6	Operational . . . . .	14
5.7	Security . . . . .	15
5.8	Legal . . . . .	15
<b>6</b>	<b>Open Issues</b>	<b>16</b>

## List of Tables

1	Revision History . . . . .	ii
---	----------------------------	----

## List of Figures

Table 1: **Revision History**

Date	Version	Notes
October 30, 2018	1.0	Created document, REV-0

# 1 Introduction

This document outlines the software requirements for Carbon Charts. The purpose of this document is to specify requirements and provide a complete description of the library to be developed. It will outline the intended purpose, features of the library, behavior, and general factors that affect its requirements. Additionally, it will illustrate the constraints under which the library must operate and conform to. This document will serve as a complete declaration of the data visualization library in development and is intended for stakeholders and the library developers, Group 18. The audience includes the stakeholders, Dr. Wassying, Nicholas Annable, Joshua Barkovic, Spencer Deevy, Viktor Smirnov and any potential developers.

## 2 Project Drivers

### 2.1 Background of the Project Effort

The information age had led to a rapid increase in the amount of raw data available. This increase in raw data is only beneficial when we use methods to derive insights from it. Humans are visual creatures, detecting patterns in sets of data is more easily interpreted using data visualization techniques. A common and convenient mode of consuming visualized data is through graphical charts. The process of mapping information to visuals is more efficient and accurate when employing the power of computing. The increased speed enables the use of much larger data sets while allowing for quick re-rendering of alternative mappings. The added benefits through computing drove application developers and organizations such as IBM Canada to highlight the need for a consistent data visualization framework.

### 2.2 Scope

Carbon Charts is an open-source JavaScript data visualization library that will be used as a functional component of IBM's Carbon Design System. The library is expected to comply with the general IBM design guidelines with the intended goal of making any data visualization processes within their products more efficient while eliminating duplicated work within different teams. It will be used to extend the data visualization capabilities of IBM Carbon

Design System by serving as an interactive, lightweight charting library. Additionally, it should be designed in accordance with design principles that follow its open source nature.

### **2.2.1 Benefits**

The proposed library merits the following benefits:

- **Reuseability:** The Carbon Charts library will allow for the use of its existing (stable and verified) visualization techniques within the software development process of other internal IBM products,
- **Extensibility:** As an open-source project, the library is easily extensible and provides for change while minimizing the impact to existing system functions. High cohesion and loose coupling allow for continual development with-in the open-source community.

## **2.3 The Stakeholders**

### **2.3.1 IBM Canada Ltd**

- IBM Canada is generally interested in disruptions within the open-source community, they invest significant amounts of effort and funds into creating an inspirational footprint within the community.
- Carbon the new IBM Design Language will define and guide everything designed by IBM, including software products, digital and traditional marketing, hardware, advertising. and more.

### **2.3.2 Application Developers**

- Web application developers can use the library to create interactive visualizations of data they would like to present to their users.
- Generally, developers would be looking for a solution that is lightweight, high-performance, reliable and accurate in results. Depending on what organization or project they are involved in, they might also have accessibility and internationalization requirements.

### 2.3.3 Application Users

- Users of web applications represent the target audience of this library. They can use charts generated by the developer and through the library, to monitor data (sensitive data in many cases), analyze patterns between data sets, make decisions based on historic representations and values, all while being able to take advantage of real-time visualizations in some cases (e.g. trading).
- Users will generally be looking for reliable functionality, an appealing visual representation and accessibility.

## 2.4 Naming Conventions and Terminology

- **Axis Chart:** Axis charts display data between a set of axis. Axis charts are 2 or 3 dimensional, with at least an x and y axis.
- **Non-Axis Chart:** Axis charts display data within an area not defined by axis. (e.g. Pie Charts).
- **Right-to-Left Support:** The ability to display languages that read right to left, such as Farsi.
- **Browser Tiers:** We define two tiers of browsers A and B. Tier A browsers consist of the two latest versions of Chrome, Firefox, Opera and Safari at any time, and any versions below those including IE-11+ will be part of tier B.
- **Test Data:** A set of example data. This set will initially consist of a random sampling of all valid data within the expected range, but will be extended to any particular data that cause issues in the software as it develops. Requirements that involve the test data in their fit criterion will be considered unmet if any member of the test data fail that requirement.
- **Demonstration Environment:** An environment resembling a common household room; no extreme conditions (temperature, pressure, magnetic fields, impact forces etc).

## 2.5 References

1. Murray, Scott. Interactive Data Visualization for the Web. O'Reilly, 2013.
2. D. L. Parnas, P. C. Clements, A rational design process: How and why to fake it, IEEE Transactions on Software Engineering, v.12 n.2, p.251-257, Feb. 1986

## 2.6 Constraints

### 2.6.1 Solution Constraints

- Must adhere to IBM open source guideline (eg. code styling, structure).
- Must operate as a component within IBM's Carbon Design System.

### 2.6.2 Schedule Constraints

- Milestone deadlines for McMaster Capstone project.
- Release deadlines for external IBM projects that rely on stable versions of Carbon Charts. These deadlines may be announced at any time in the future.
- Due to team members having other obligations, the project is limited to 10 hours per week. Team members may opt to exceed this limit, but it is not required.

### 2.6.3 Budget Constraints

- \$750 CAD is the budget constraint imposed on us by the assignment.

## 3 Off-The-Shelf Solutions

- d3.js is a widely used JavaScript library for data visualization. Initially released in 2011, it has gone through many levels of user testing, bug discoveries and refactors, and has iterated to become an industry standard for web visualizations. The library offers some core functionality to be able to select and manipulate SVG elements, implement quick

and easy progressive web transitions, perform data-binding on DOM elements etc.

- Chart.js is an open source JavaScript charting library for designers and developers.
- Chartist.js an open source JavaScript charting library. It is the product of a community that was disappointed about the abilities provided by other charting libraries.
- (Hardware) Arduino is an open-source electronic prototyping platform enabling users to create interactive electronic objects. It can be used to interface with various sensors.

## **3.1 Relevant Facts and Assumptions**

### **3.1.1 Facts**

- Data representations are often only defined for specific data types. For example, it is obvious that charts with numeric axis can only accept numerical values on that axis. It will always be assumed that any such invalid data is not present.
- Data sets are unbounded in size and can have any number of elements. We will assume an "expected data range" of 0 to 50 data points. Data sets larger than this cannot be usefully displayed in a visual graph due to screen space limitations. This is a strict limit which allows for application developers using this library to define how large sets of data are mapped.
- The hardware will only be expected to operate in the demonstration environment.

### **3.1.2 Assumptions**

- It is assumed that developers will use library functions correctly, and pass in the correct data types within the correct data range.



## 3.2 Context Diagrams

The following context diagram defines the boundaries of the system and the entities that act with it.

# 4 Functional

## 4.1 IBM and Library Requirements

### Requirement #: FN-1

**Description:** The developer can render a list of data and a list of values as a chart.

**Rational:** Basic functionality.

**Fit Criterion:** For any chart type, the test data can be rendered.

**Stakeholders:** Application Developers, IBM Canada Ltd

### Requirement #: FN-2

**Description:** The developer can choose the type of chart to be displayed.

**Rational:** Basic functionality.

**Fit Criterion:** For all charts types, the test data can be rendered.

**Stakeholders:** Application Developers, IBM Canada Ltd

### Requirement #: FN-3

**Description:** Subsection colours must match the colour corresponding to that subsection legend entry.

**Rational:** Legend entries must be visually connected to their data without requiring text.

**Fit Criterion:** Visual confirmation of rendered test data and legend.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

### Requirement #: FN-4

**Description:** All data points in axis charts must be directly vertical of the x-axis tick correlating to that point's label and directly horizontal of the y-axis tick corresponding to that point's value.

**Rational:** This allows data to be immediately correlated with it's corresponding axis values.

**Fit Criterion:** Visual confirmation of rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

**Requirement #: FN-5**

**Description:** The user has the option to save a graph as an SVG image.

**Rational:** Allows the user to export the graph without requiring them to use a screenshot tool.

**Fit Criterion:** SVG image is visually similar to rendered graph.

**Stakeholders:** Application Users

**Requirement #: FN-6**

**Description:** Library must respect The right-to-left preference in a user's browser and display text right-to-left.

**Rational:** Not all users read left-to-right.

**Fit Criterion:** All other requirements are met when rendering text right-to-left.

**Stakeholders:** Application Users

**Requirement #: FN-7**

**Description:** The developer shall be able to specify a rectangular region behind an axis chart can be displayed.

**Rational:** This is a common way to indicate a special region of the data to the user.

**Fit Criterion:** Visual confirmation of rendered test data with a threshold.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

**Requirement #: FN-8**

**Description:** The user can adjust the graph viewpoints, making data larger or smaller and changing the visible region. Axis and legends should remain where they are.

**Rational:** Many data sets have a variety of scale and need to be explored from multiple viewpoints.

**Fit Criterion:** The user can zoom and scroll the rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

**Requirement #: FN-9**

**Description:** The developer can disable any visual elements of the software for their application.

**Rational:** Some elements may be unnecessary or unwanted, depending on the application.

**Fit Criterion:** Given any visual element, it can be removed by the developer.

**Stakeholders:** Application Developers

#### **Requirement #: FN-10**

**Description:** The users ability to resize any chart should be limited to the minimum and maximum scale size of a chart.

**Rational:** Developers may want to enforce a viewing scale that they believe is appropriate for the data.

**Fit Criterion:** If a minimum or maximum scale size limit is enabled on a chart rendering the test data, a user cannot scale past that limit.

**Stakeholders:** Application Developers

#### **Requirement #: FN-11**

**Description:** Transitions made by selecting legend items should correctly adjust the chart, load data and generate elements using the selected data sets.

**Rational:** Users should not lose any data due to chart transitions. This is a common problem in charting applications.

**Fit Criterion:** All test data is still visible after a transition, unless it was explicitly hidden.

**Stakeholders:** Application Users, Application Developers

## **4.2 Bar Chart : Stack and Grouped**

#### **Requirement #: FN-12**

**Description:** Data can be rendered as rectangular subsections which originate from an x axis, and stretch upward.

**Rational:** Definition of a correct bar chart.

**Fit Criterion:** Visual confirmation of rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

#### **Requirement #: FN-13**

**Description:** Data can be rendered as multiple series of stacked rectangular subsections which originate from an x axis, and stretch upward.  
**Rational:** Definition of a correct stacked bar chart.  
**Fit Criterion:** Visual confirmation of rendered test data.  
**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

### 4.3 Circle Charts : Pie, Donut and Bubble

#### Requirement #: FN-14

**Description:** Data can be rendered as a circle is divided into subsections which stretch from the outer radius to the center.(this is a separate requirement) Each subsection is assigned a different colour.  
**Rational:** Definition of a correct pie chart.  
**Fit Criterion:** Visual confirmation of rendered test data.  
**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

#### Requirement #: FN-15

**Description:** Data can be rendered as a circle is divided into subsections which stretch from border to center. Each subsection is assigned a different colour.  
**Rational:** Definition of a correct donut chart.  
**Fit Criterion:** Visual confirmation of rendered test data.  
**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

#### 4.3.1 Pie and Donut Charts

#### Requirement #: FN-16

**Description:** A perpendicular X and Y axis is displayed, unless the developer explicitly excludes this feature or the chart is a pie or donut chart.  
**Rational:** Definition of a correct axis chart.  
**Fit Criterion:** Visual confirmation of rendered test data and axis.  
**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

#### Requirement #: FN-17

**Description:** The maximum number of subsections in a pie or donut chart is 7. If the number of subsections exceeds this, the smallest subsections will be collapsed into an "other" subsection until there are only 7 subsections.

**Rational:** It is desirable to simplify these chart types.

**Fit Criterion:** Visual confirmation of rendered test data with  $\leq 7$ , 7, and  $\geq 7$  subsections.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

#### **Requirement #: FN-18**

**Description:** Each data point in a pie or donut chart is displayed with an area proportional to that data point's absolute value divided by the sum of all data points.

**Rational:** Absolute data should be renormalized in these chart types to fit the available display area.

**Fit Criterion:** Visual confirmation of rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

### **4.3.2 Bubble Charts**

#### **Requirement #: FN-19**

**Description:** Bubble Charts must consist of ovular or circular subsections within the boundaries of the x and y axis. Individual subsections must be one consistent colour.

**Rational:** Definition of a correct bubble chart.

**Fit Criterion:** Visual confirmation of rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

### **4.4 Line Chart**

#### **Requirement #: FN-20**

**Description:** Data can be rendered as coordinates plotted on an x and y grid, connected by a line.

**Rational:** Definition of a correct line chart.

**Fit Criterion:** Visual confirmation of rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

## 4.5 Area Chart

**Requirement #:** FN-21

**Description:** Area chart subsections must be within the boundaries of the x and y axis. Individual subsections must be one consistent colour.

**Rational:** Definition of a correct area chart.

**Fit Criterion:** Visual confirmation of rendered test data.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

## 4.6 Hardware Requirements

**Requirement #:** FN-22 (Hardware)

**Description:** The hardware component records real-time sensory data, including temperature, pressure, and proximity, and is displayed using the charts.

**Fit Criterion:** Data recorded by the sensors accurately reflects the demonstration environment.

**Stakeholders:** Application Users

**Requirement #:** FN-23 (Hardware)

**Description:** The hardware component can transfer data to the display device and the data is displayed using the software's charts.

**Fit Criterion:** Demonstration data from the hardware component is accurately displayed in a chart.

**Stakeholders:** Application Users

# 5 Non-Functional

## 5.1 Look and Feel

**Requirement #:** LF-1

**Description:** All animations must be rendered smoothly.

**Rational:** Jittery animations can interfere with data interpretation.

**Fit Criterion:** Visual affirmation of rendered and animated test data.

**Stakeholders:** Application Users

**Requirement #:** LF-2

**Description:** Color Maps or the palettes used should adhere to IBM design color specs separating palettes by standard qualitative and high contrast qualitative.

**Rational:** IBM Design guidelines and defined coloring principles.

**Fit Criterion:** Visual affirmation of rendered test data.

**Stakeholders:** Application Developers and Users

#### **Requirement #: LF-3**

**Description:** X-axis must always be positioned at the bottom of the chart with the labels rotated  $-45^\circ$  by default and aligned to the right of the y-axis.

**Rational:** IBM Design guidelines and defined coloring principles.

**Fit Criterion:** Visual affirmation of rendered test data.

**Stakeholders:** Application Developers and Users

#### **Requirement #: LF-4**

**Description:** Y-axis must display no more than 6-digit metrics on a single axis. Separate multiple metric title labels with commas.

**Rational:** IBM Design guidelines and defined coloring principles.

**Fit Criterion:** Visual affirmation of rendered test data.

**Stakeholders:** Application Developers and Users

## **5.2 Usability**

#### **Requirement #: USB-1**

**Description:** Using a screen-reader, one should be able to navigate across all elements of a chart, and hear back information regarding labels and values corresponding to the element.

**Rational:** Carbon Charts should be usable by the widest possible range of users, including visually impaired individuals.

**Fit Criterion:** The rendered test data can be fully interpreted using JAWS screenreader.

**Stakeholders:** Application Users, IBM Canada Ltd

#### **Requirement #: USB-2**

**Description:** Charts and legends should be able to render using pattern fills rather than solid color fills in accessibility mode.

**Rational:** Visual accessibility is also important for visually impaired individuals, including individuals with colour blindness. Legend items should also follow the same logic.

**Fit Criterion:** The rendered test data can be fully interpreted in black-and-white.

**Stakeholders:** Application Users, IBM Canada Ltd

### 5.3 Safety

Developers using this library should independently evaluate it's properties and how they relate to their system's safety requirements. Due to this software's intended use as a module within another application, it is not possible for us to provide safety guarantees without knowing the application's specific requirements.

We will make efforts to ensure documentation of the software is up-to-date and accurate to allow developers to make informed decisions on the suitability of this software for their application, in regards to safety or otherwise.

### 5.4 Performance

**Requirement #: PERF-1**

**Description:** The compiled bundle size must be of a reasonable size.

**Rational:** The bundle should remain small to ensure quick load times.

**Fit Criterion:** The compiled bundle size must be less than 150kb, including compression.

**Stakeholders:** Application Users

**Requirement #: PERF-2**

**Description:** The render time of charts should be short.

**Rational:** The user will not want to wait long times for data to render.

**Fit Criterion:** The render time of a chart inside the expected data range should be 50ms without animations, and 800ms with animations.

**Stakeholders:** Application Users

**Requirement #: PERF-2**

**Description:** The software should not use a large amount of volatile memory.

**Rational:** Other applications running on the same page should not struggle with memory shortages due to the software.

**Fit Criterion:** The volatile memory footprint of the software does not exceed 3mB while rendering the test data.



**Stakeholders:** Application Users

**Requirement #:** PERF-3 (Hardware)

**Description:** The hardware component records data values at a frequency of at least  $100ms^{-1}$ .

**Rational:** The data should be refreshed often enough that two chart refresh cycles do not pass without new data.

**Stakeholders:** Application Users

## 5.5 Installability

**Requirement #:** INST-1

**Description:** The software is installable with Node Package Manager.

**Rational:** Other applications running on the same page should not struggle with memory shortages due to the software.

**Fit Criterion:** The command `npm install carbon-charts` is sufficient to install the software.

**Stakeholders:** Application Developers

## 5.6 Operational

**Requirement #:** OP-1

**Description:** The software is supported by any operating system that supports Node.js.

**Rational:** Developers should not be limited by their operating system when deciding to use the software.

**Fit Criterion:** No operating system with at least %2 market share is unsupported.

**Stakeholders:** Application Developers

**Requirement #:** OP-2

**Description:** The software is fully supported by all Tier A browsers.

**Rational:** Users with common browsers should not encounter issues.

**Fit Criterion:** None of the defined Tier A browsers are unsupported.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

**Requirement #:** OP-3

**Description:** The software is partially supported by all Tier B browsers.  
**Rational:** An attempt should be made to support fallbacks for uncommon browsers in order to avoid excluding users.  
**Fit Criterion:** All Tier B browsers can at least display the test data.  
**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

## 5.7 Security

### Requirement #: SEC-1

**Description:** The software must not utilize any deprecated or insecure dependencies.

**Rational:** The security of the software should not be compromised by a dependency.

**Fit Criterion:** No warnings should be present when the software is scanned using NPM's Node Security Platform.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

### Requirement #: SEC-2

**Description:** The software must not make any external HTTP requests.

**Rational:** This reduces the cross-site scripting (XSS) attack surface.

**Fit Criterion:** No form of HTTP request code should be present in the software.

**Stakeholders:** Application Users, Application Developers, IBM Canada Ltd

## 5.8 Legal

### Requirement #: LGL-2

**Description:** The software must meet IBM's arbitrary legal requirements.

**Rational:** IBM should be able to use the software without encountering legal issues.

**Fit Criterion:** Review and approval of the final software by IBM's legal process.

**Stakeholders:** IBM Canada Ltd

### Requirement #: LGL-2

**Description:** The software must conform to IBM's Open Source Guidelines.

**Rational:** IBM should be able to use the software without encountering conflicts with their guidelines.

**Fit Criterion:** Review and approval of the final software by IBM.

**Stakeholders:** IBM Canada Ltd

**Requirement #: LGL-3**

**Description:** All dependencies and imported code must be compatible with the MIT License.

**Rational:** The software is to be licensed as MIT.

**Fit Criterion:** All used code has a MIT-compatible license.

**Stakeholders:** Application Developers, IBM Canada Ltd

## 6 Open Issues

N/A